# WAY TO SIGNIEL

## 머드게임제작 발표

AR 인공지능 게임 프로그램 개발
한창신

# CONTENTS



# 01 게임소개

1. 게임소개
2. 진행방식

# 1. 게임소개

1. 게임소개

   제목: WAY TO SIGNIEL

   내용: 경마에 승리해서 롯데 시그니엘로 이사가자는 의미.

   소개: 주어진 자본으로 경마에 베팅 하고 경마 경기를 시청 후 경기 결과에 따라 배당금 수취.

2. 진행방식

   베팅: 플레이어는 보유한 금액 안에서 자유롭게 베팅. (중복 가능)

   각 말은 한번 이동시 1/3/5칸 중 이동하며 확률과 배당률 상이.

   아이템: 플레이어는 아이템을 구매하여 원하는 말의 이동 확률을 조정 가능. (중복불가)

   초기화: 베팅 후 경기를 시작하지 않았다면 베팅한 금액 초기화 기능.

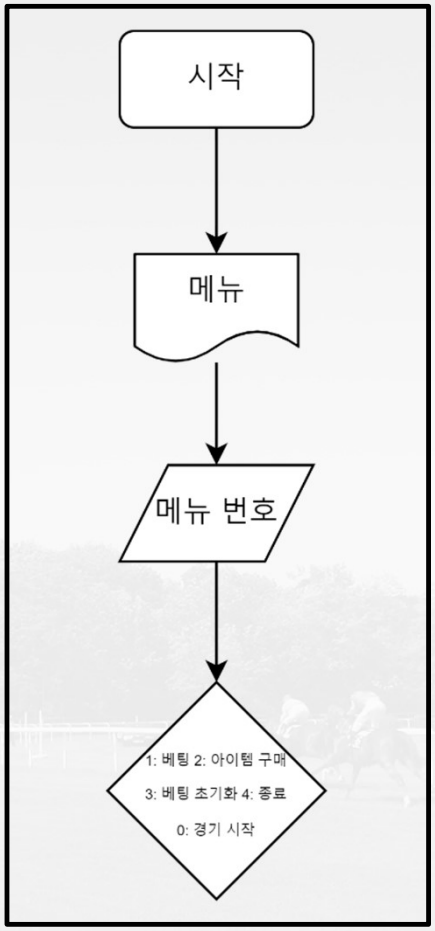
   경기 시작: 경마가 시작되면 경기가 약 10초간 진행되며, 경기 후 결과에 따라 배당금 수취.

## 02 간략 순서도

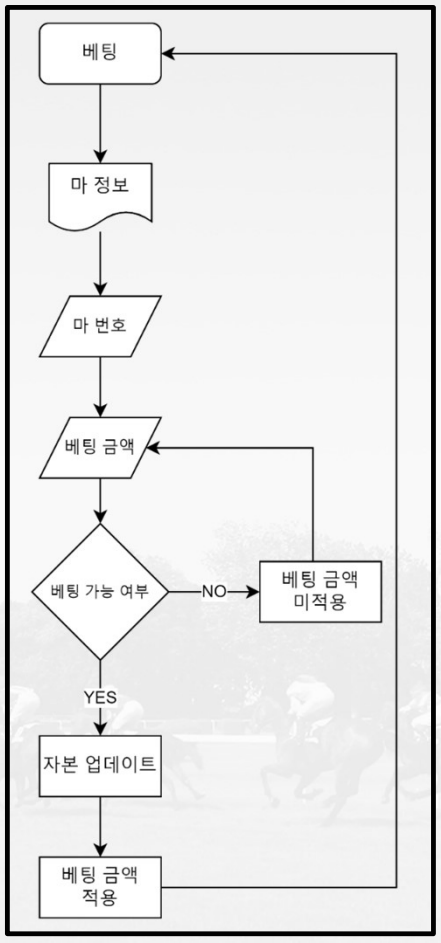
1. 시작
2. 베팅
3. 아이템 구매
4. 초기화
5. 종료
6. 경기

# 2. 간략 순서도

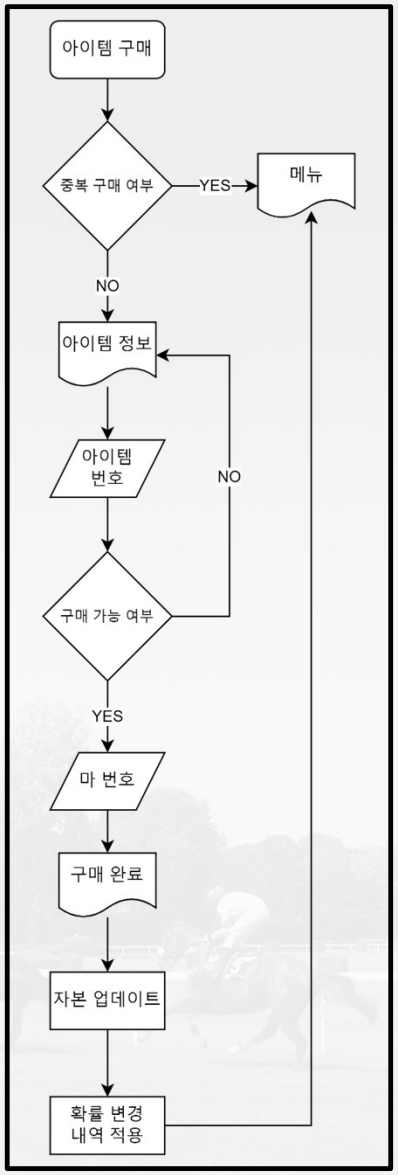
## 2-1. 시작 순서도



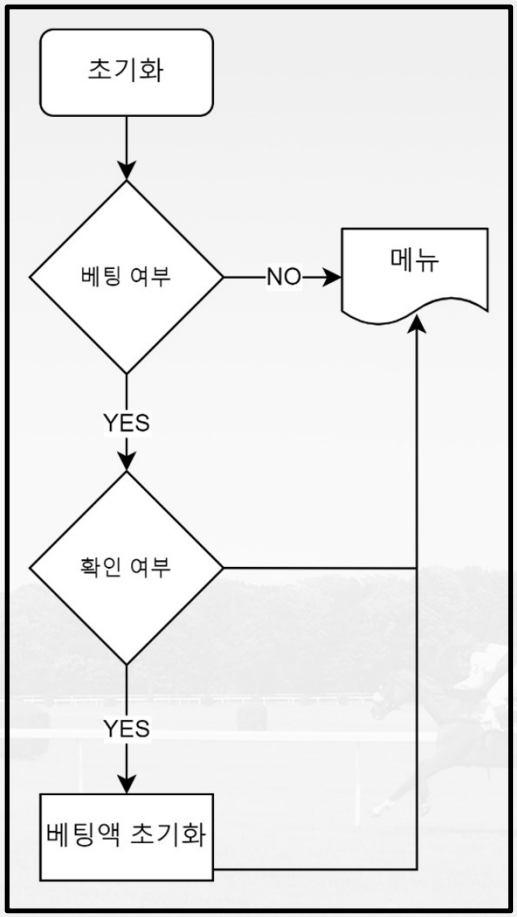

## 2-2. 베팅 순서도




## 2-3. 아이템 구매 순서도

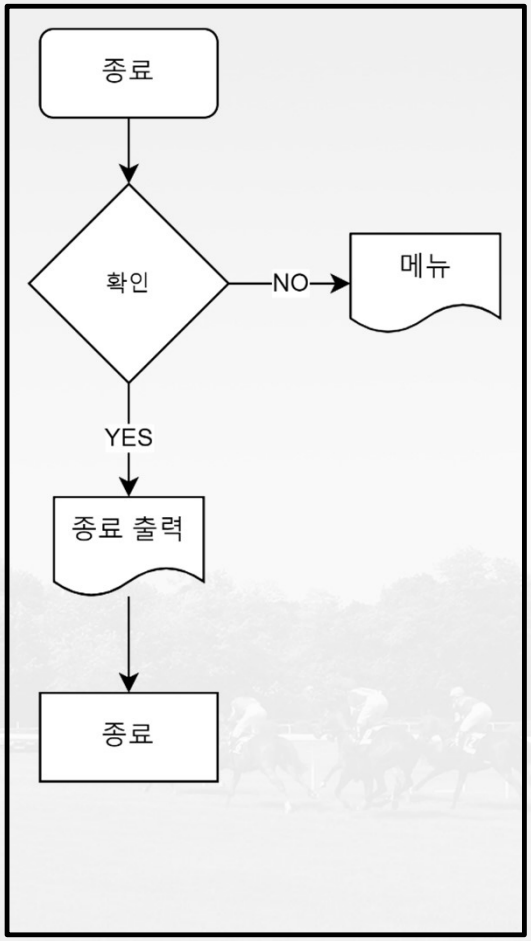



## 2-4. 초기화 순서도

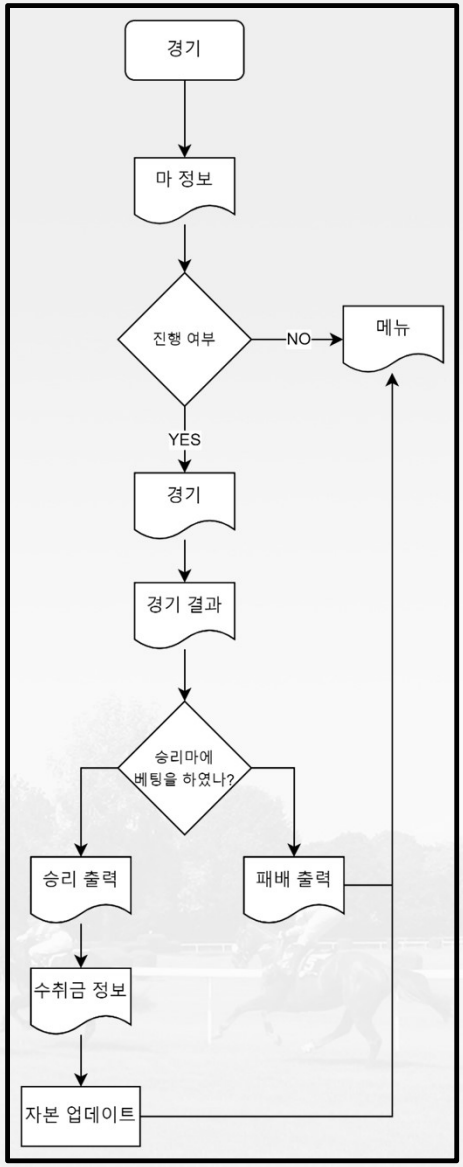


# 2. 간략 순서도

2-5. 종료 순서도




2-6. 경기 순서도



# 03 게임 실행

## 04 주요 함수

1. 오프닝 함수
2. 입력 함수
3. 문자씩 나오는 함수 (+서식지정자)
4. 말 이동 확률 함수
5. 말 이동 함수

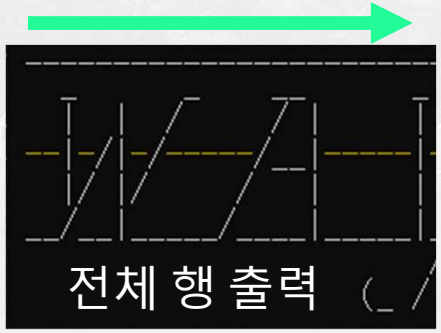
# 4. 주요 함수

1. 오프닝 함수

   - char str[8][100]에 파일의 8줄을 입력 받는다.

   - 전체 행을 기준으로 n만큼의 열을 뽑는다.

   - n을 뽑고 나서 삭제하고 0부터 n+1을 재출력을 반복한다.




0 →0~1 →0~2...
0~20 →....0~끝

삭제 후 출력 반복

```
void printstart()
{
    FILE* fp;
    char str[8][100] = { 0 };

    play_music("opening", 1);

    fp = fopen("text.txt", "r");
    if (fp == 0)
    {
        printf("text.txt 열기 실패");
        exit(1);
    }

    system("cls");
        for (int i = 0; i < 8; i++) //str배열에 8줄 문자열 정의
        {
            fgets(str[i], 100, fp);
        }

        for (int i = 0; i < 100; i++) //총 열의 개수
        {
            for (int k = 0; k < 8; k++) //총 행의 개수
            {
                for (int j = 0; j < i; j++) //출력할 행의 개수 ==> 전체 행을 0 -> 0~1 -> 0~2 -> 0~3... 출력
                {
                    if (k == 3 && str[k][j] == '-') printf("₩033[33m%c", str[k][j]);
                    else printf("₩033[0m%c", str[k][j]);

                }
                if (i < strlen(str[k])) DOWN;
            }
```

# 4. 주요 함수

2. 입력 함수

- operation을 입력 받기 위해 생성한 함수.
- 입력 받을 때 마다 받을 수 있는 숫자가 상이하여 최소값과 최댓값을 입력 받고, 범위 내로 입력을 받았다면 입력한 번호를 반환하는 함수.
- 입력 받을 때 마다 번호 확인 및 while문을 작성하지 않아도 되어서 편리.

```
//입력 함수
int input(int min, int max)
{
    int operation;

    printf("₩n");
    while (1)
    {
        printf("₩t입력: ");

        scanf("%d", &operation);

        if (operation >= min && operation <= max)
        {
            DOWN;
            break;
        }
        else
        {
            DOWN;
            printf("                                    ₩r");
        }
    }

    return operation;
}
```

# 4. 주요 함수

3. 한 문자씩 나오는 함수 (+서식지정자)

   - 문자열을 일정한 시간 간격을 두고 한 문자씩 출력되는 함수.

   - 문자열을 매개변수로 입력받은 후 char str[500]에 대입 후 각 배열을 시간간격을 두고 출력.

   - printf("문자열")과 같이 printw("문자열")과 같이 편리하게 사용 가능.

```
void printw(char* input)
{
    char str[500];
    strcpy(str, input);

    for (int i = 0; i < strlen(str); i++)
    {
        putchar(str[i]);
        Sleep(WORD_SPEED);
    }
}
```

# 4. 주요 함수

3. 한 문자씩 나오는 함수 (+서식지정자)

   - 문자열을 일정한 시간 간격을 두고 한 문자씩 출력되는 함수.

   - 문자열을 매개변수로 입력받은 후 char str[500]에 대입 후 각 배열을 시간간격을 두고 출력.

   - printf("문자열")과 같이 printw("문자열")과 같이 편리하게 사용 가능.

```
void printw(char* input)
{
    char str[500];
    strcpy(str, input);

    for (int i = 0; i < strlen(str); i++)
    {
        putchar(str[i]);
        Sleep(WORD_SPEED);
    }
}
```

```
==========================================================================================
진심으로 축하드립니다.!!
당신은 5번마 ◐우리 우디◐에 0억 1000만원을 베팅 하셨고, 배당률은 10 배 입니다.
총 수령 금액은 1억 0 만원 입니다. 다시 한번 진심으로 축하드립니다!!!

                                                                press enter
==========================================================================================
```

# 4. 주요 함수

3. 한 문자씩 나오는 함수 (+서식지정자)

   - printw를 사용하게 되면 상황에 따라 변경되는 서식지정자 출력 불가.

   - 서식 지정자가 필요한 부분의 위치를 확인해서 각 위치마다 필요한 정보를 별개로 출력

   - 예를 들어, 해당 문자열의 33번째에 승리 말의 번호가 들어가야 한다면, 0~32번째는 str[i]를 출력하고, i=33 때는 printf("%d",horse.num)을 출력.

```
void if_wins(struct horses horse)
{

    //32: 마 번호, 38: 마 이름, 42: 베팅액 73: 배당률 98: 수
    char str[300] = "진심으로 축하드립니다.!! ₩n₩
당신은  번마 에 억 만원₩033[0m을 베팅 하셨고, 배당률은  배 입
총 수령 금액은 억  만원₩033[0m 입니다. 다시 한번 진심으로 축

    for (int i = 0; i < strlen(str); i++)
    {
        if (i == 33) str[i] = horse.num + '0';
        if (i == 39) printf("₩033[33m%s%s%s₩033[0m", horse.in
        if (i == 42) printf("₩033[33m%d", get_uk(horse.tot_be
        if (i == 45) printf("%d", get_man(horse.tot_bet));
        if (i == 78) printf("₩033[33m%d₩033[0m", horse.percen
        if (i == 105) printf("₩033[33m%d", get_uk(horse.tot_b
        if (i == 108) printf("%d", get_man(horse.tot_bet * ho
        putchar(str[i]);
        Sleep(WORD_SPEED);
    }
}
```

베팅을 맞췄을 경우 나오는 문자열.

해당 문자열에는 승리 마와 베팅 액수, 배당률, 수취금 등이 포함되어 있다.

# 4. 주요 함수

4. 말 이동 확률 계산

- 말 구조체 선언

```
//마 구조체 선언
struct horses {
    int num;
    char* name;
    char imo[5];      //경마 시뮬레이
    int per_1;        //N칸 갈 확률
    int per_3;
    int per_5;
    int tot_steps;
    int percent;      //배당율
    int bet;
    int tot_bet;
};
```

- 말 정보입력

(마지막 4번째 ~ 2번째 가 1/3/5칸 이동 확률 (%))

```
void reset_horses(struct horses* horse)
{
    set_horses(&horse[0], 1, "이상해씨", "☎", 50, 25, 25, 2);
    set_horses(&horse[1], 2, "키타산블랙", "※", 55, 20, 25, 4);
    set_horses(&horse[2], 3, "까미 냥이", "◎", 60, 20, 20, 6);
    set_horses(&horse[3], 4, "CY.NIRO ", "◈", 65, 15, 20, 8);
    if (GM == 1)set_horses(&horse[4], 5, "우리 우디", "◐", 70, 15, 15, 10);
    else set_horses(&horse[4], 5, "우리 우디", "◐", 0, 0, 100, 10);
}
```

# 4. 주요 함수

4. 말 이동 확률 계산

- 0~99까지 랜덤으로 숫자를 뽑는다.
- 숫자가 1칸 갈 확률보다 낮으면 1칸을 반환.
- 숫자가 1칸 확률 보다 높지만, 1칸+3칸 확률보다 낮으면 3 반환.
- 숫자가 1칸+3칸 확률보다 높지만, 1칸+3칸+5칸 확률(=99) 보다 낮으면 5칸.

```
int rand_step(struct horses horse)
{
    int dice;

    dice = rand() % 100;
    horse.per_1 -= 1; //0부터이기 위해
    if (dice <= horse.per_1) return 1;
    else if (dice <= horse.per_1 + horse.per_3) return 3;
    else if (dice <= horse.per_1 + horse.per_3 + horse.per_5) return 5;
}
```

# 4. 주요 함수

5. 말 이동 함수

- 3칸이나 5칸 이동 할 때 한번에 이동하는 것이 아닌 한 칸 씩 3/5번 이동.
- 1/3/5칸 상관없이 소요시간 동일.
- STEP(이동 칸 수) 만큼 반복문을 진행하고, 이동하는 시간을 STEP만큼 나누었다.

ex)

1칸: 한칸 이동 후 SEC_STEP sleep.

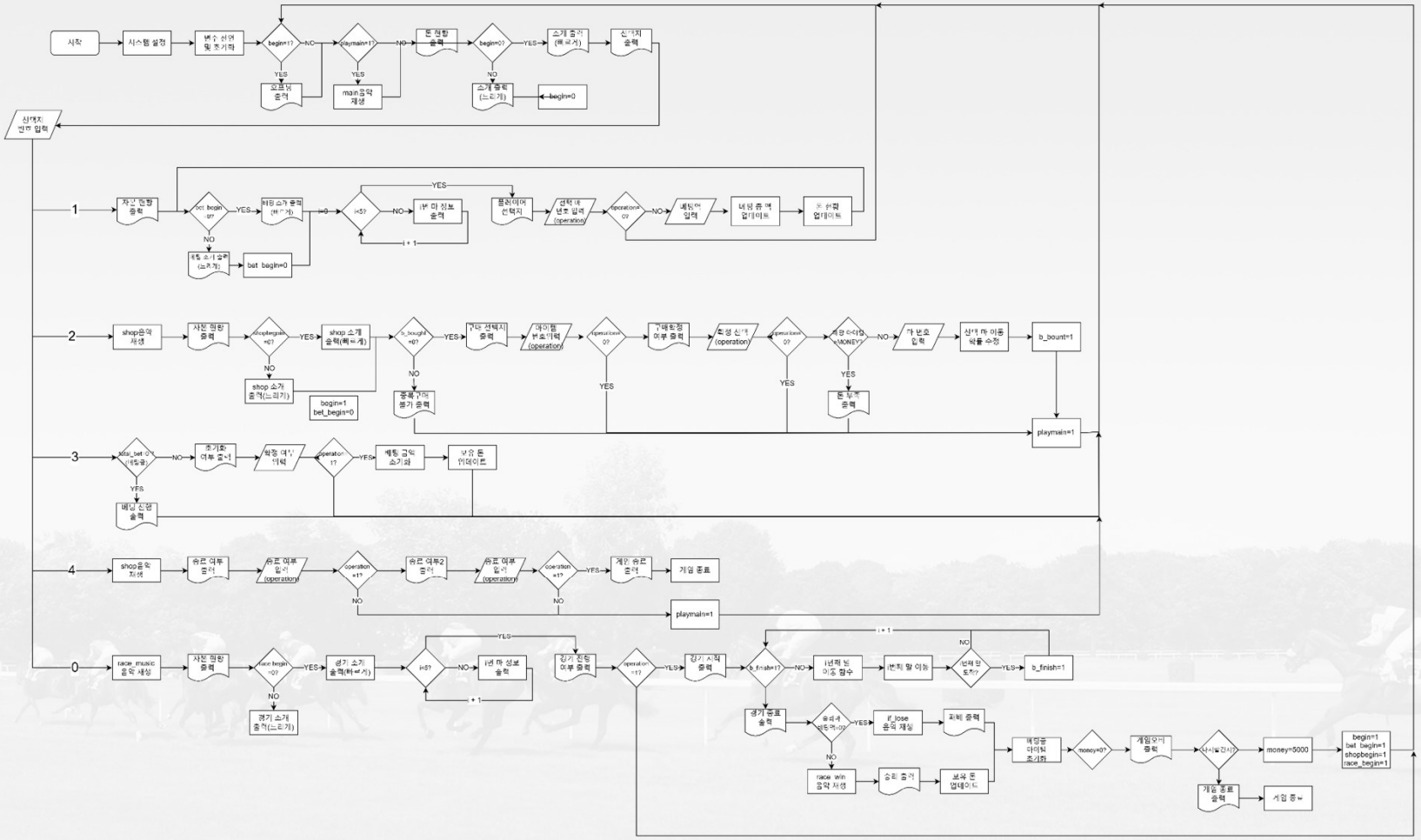
3칸: 한칸씩 3번 SEC_STEP/3 속도로 반복


```
for (int k = 0; k < step; k++) //step 만큼 반복: 삭제->출력
{
    if (k != 0)
    {
        //한줄 삭제하기 (특수문자는 칸을 2칸 차지하나봄 ㅎ)
        for (int i = 0; i < DIST; i++) printf("₩b₩b");
    }

    for (int i = 0; i < DIST; i++) //한줄 만들기
    {
        if (horse->tot_steps + go == i) printf("₩033[32m%s", horse->imo);
        else if (i < horse->tot_steps + go) PASSED;
        else PATH;

        if (horse->tot_steps + go >= DIST - 1)
        {
            *finish = 1;
            *winner = horse->num-1;
        }
    }
    go++;
    if (*winner != 100) break;

    Sleep(SEC_STEP / step); //움직이는 속도 조정 (몇 칸을 가든 소모되는 시간
}
printf("₩n₩n");
horse->tot_steps += step;
```

# 05 MAIN 순서도

1. 순서도

# 5. 순서도


시작
시스템 설정
변수 선언 및 초기화
begin==1?
YES
오프닝 출력
NO
playmain==1?
YES
main음악 재생
NO
돈 현황 출력
begin==0?
NO
소개 종료 (느리게)
begin=0
YES
소개 출력 (빠르게)
선택지 출력
선택지 번호 입력
1
자산 현황 출력
bet_begin==0?
YES
처음 소개 출력 (빠르게)
NO
베팅 소개 출력 (느리게)
bet_begin=0
i=0
i<5?
NO
i번 마 정보 출력
i + 1
YES
플레이어 선택지
선택 번호 입력 (operation)
operation==0?
NO
베팅액 입력
배팅 금액 업데이트
보유 돈 업데이트
2
shop음악 재생
자본 현황 출력
shopbegin==0?
YES
shop 소개 출력(빠르게)
NO
shop 소개 출력(느리게)
begin=1
bet_begin=0
b_bought==0?
YES
NO
종료 가능 불가 출력
기능 선택지 출력
아이템 번호 입력 (operation)
operation==0?
YES
구매 확정 여부 출력
확정 선택 (operation)
operation==0?
YES
보유 돈<MONEY?
NO
YES
돈 부족 출력
가 번호 입력
선택 마 이동 칸을 수정
b_bount=1
playmain=1
3
total_bet==0?
YES
NO
베팅 진행 출력
초기화 여부 출력
확정 여부 입력
operation==0?
YES
NO
베팅 금액 초기화
보유 돈 업데이트
4
shop음악 재생
슈르 여부 출력
슈르 여부 입력 (operation)
operation==0?
NO
슈르 여부2 출력
슈르 여부 입력 (operation)
operation==0?
YES
NO
계산 슈르 출력
게임 종료
playmain=1
0
race_music 음악 재생
자본 현황 출력
race begin==0?
YES
NO
경기 소개 출력(하나씩)
경기 소개 출력(느리게)
i=0?
YES
NO
i번 마 정보 출력
i + 1
경기 진행 여부 출력
operation==1?
YES
경기 시작 출력
b_finish==1?
NO
i + 1
i번째 돈 이동 함수
i번째 말 이동
NO
i번째 칸 도착?
YES
b_finish=1
경기 종료 출력
승리자 판독==0?
YES
if_lose 음악 재생
패배 출력
NO
race win 음악 재생
승리 출력
보유 돈 업데이트
배당금 다이빙 초기화
money==0?
게임오버 출력
시작할건가요?
YES
money=5000
begin=1
bet_begin=1
shopbegin=1
race_begin=1
게임 종료 출력
NO
시업 중ㅠ

# 감사합니다.